

Filtro de autorización

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a nuestro curso sobre servlet. En el video pasado puede ser que a ustedes no les haya quedado muy, muy en claro lo que es un filtro, pero una cosa que sí les ha quedado en claro seguro es que nosotros no hemos tocado las acciones, no hemos colocado ese control, ese monitoreo del tiempo en cada una de nuestras acciones, sino que lo hemos dejado fuera de un filtro y en un filtro.

[00:33] Y así como lo hemos colocado en un filtro, nosotros podremos crear un segundo filtro, también, podemos tener un segundo filtro en el cuál, ahí por ejemplo, pongamos la autorización para ver si, perdón autenticación, a ver si la persona realmente está lograda o no.

[00:52] Y digamos ese código para saber si una persona está logada podríamos colocarlo en ese filtro y sacarlo de nuestro controlador para que nuestro controlador quede con una única funcionalidad, controlar a ver cuál acción tiene que llamar.

[01:05] Entonces vamos a intentar hacer eso. Vamos a ver si queda un poco más claro también el tema del filtro al realizar un segundo filtro. Bueno, nosotros tenemos acá nuestro servlet, en nuestro paquete servlet tenemos nuestro filtro de monitoreo, entonces vamos a hacer clic derecho en nuestro servletNew y esta vez vamos a usar el menú contextual de filter. Vamos a hacer eso.

[01:28] Acá en nombre de la clase le voy a colocar autorización, autorizaciónFilter, ya que le había puesto monitoreo filter en inglés filter,

entonces le vamos a dejar filter en inglés. Acá está súper clase, ustedes tienen que eliminarla, ese `httpFilter`. Vamos a colocar en next el nombre del `urlPattern` no es `autorizaciónFilter`, va a ser la misma entrada cuando la persona va a nuestra `urlEntrada` también tiene que pasar por este filtro. Next.

[02:06] Y acá, dentro de las interfaces, vamos a agregar nuestro filter de `jakarta.servlet`. Si a ustedes no les aparece ahí en la primera ventana, simplemente busquen filter acá y asegúrense de seleccionar estos `jakarta.servlet`, filter de `jakarta.servlet`, muy importante eso. Finish.

[02:25] Y acá tenemos algunas cosas para arreglar. Primero, vamos a eliminar ese comentario del filtro, vamos a abrir acá nuestros imports y vean lo siguiente. Nosotros por lo menos acá en esta versión que estoy utilizando desde Eclipse y de Tomcat, yo tengo importado los filtros de `jakarta.servlet` y los filtros de Java X. ¿Por qué está ocurriendo esto?

[02:56] Puede ser que si ustedes están viendo este video muy en el futuro, puede ser que a ustedes no les aparezca este problema y simplemente estén con `jakarta`. Lo que ha ocurrido es que a partir de Java 9, si no me equivoco. Java EE, Java Enterprise Edition pasó a ser parte de la Eclipse Foundation, la Fundación Eclipse.

[03:18] Entonces antes era de Oracle, pasó a ser ahora de Eclipse, pero se vieron obligados en Oracle a no usar más la palabra Java para ese paquete, ya que Java es una marca registrada de Oracle, entonces han tenido que cambiar a `Jakarta`. Entonces vamos a hacer lo siguiente. Vamos a sacar ese Java X, todos esos importes que incluyen Java X.

[03:45] Bien, acá van a ver que aparece duplicado ese `init`, por ejemplo aparece acá y acá al final, voy a minimizar acá esta ventana, vamos a dejarlo todo mayor para que se pueda ver un poco mejor. Tenemos dos `inits`. Tenemos ese `init` y ese `init`. Uno que tiene un `jakarta.servletException` y el otro es `servletException`, nada más.



[04:11] Como yo elimine los imports de Java X entonces no necesito más especificar cuál `ServletException` yo quiero usar, simplemente colocando `ServletException` ya debería obtener, bueno, no está pareciendo acá, pero lo voy a importar, `Java ServletException` de jakarta, ahí está.

[04:34] Importé el `ServletException` de jakarta, entonces ya no necesito más especificar cuál `ServletException` es el que tengo acá, entonces puedo eliminar esto, tenemos un constructor acá, no lo voy a usar, no lo necesitamos en la clase anterior y no lo vamos a necesitar en este, entonces lo elimino, ya nuestra implementación de filter ya sabe cuál constructor utilizar. Lo mismo con `doFilter`.

[05:00] Tenemos un segundo `doFilter` acá y no lo necesito, entonces vamos a eliminar esto, nos queda un `doFilter`, acá tenemos un `destroy` que si queremos también podríamos modificar el comportamiento, pero no, no lo necesitamos realmente, entonces podemos eliminar eso también y eliminar también ese `init`. Al final el `ServletException` sirvió para este `doFilter`, bien.

[05:26] Tenemos un import que no lo utilizamos, entonces "Ctrl + Shift + O" para eliminar. Perfecto, voy a eliminar este comentario también. Vean que hemos eliminado muchas cosas y eliminamos eso. Perfecto. Nos queda con lo mismo que teníamos en nuestro filter antes de que nosotros hiciéramos las modificaciones.

[05:49] ¿Qué más nos falta? Acá nos faltó una cosa, nos falta un `@WebFilter`, especificando cuál es nuestra `urlPattern`. Acá es `urlPatterns = "/entrada"`, "Ctrl + S". En la versión de ustedes puede ser que les ocurra la misma cosa que me ocurre a mí, doble clic en nuestro nombre de archivo para que aparezca acá nuestro explorador y en el `web.xml` les va a aparecer acá el filtro.

[06:24] Háganme caso, elimínenlo y solo para para que yo les pueda mostrar una cosa después en el final, entonces simplemente guarden "Ctrl + S", ese

web.xml, cierran y volvemos a nuestro AutorizaciónFilter. Bien. Tenemos lo mismo que teníamos en nuestro MonitoreoFilter.

[06:44] ¿Ahora qué es lo que tendríamos que hacer? La idea de esta AutorizaciónFilter era llevar nuestro comportamiento para ver si un usuario está bloqueado o no. Lo teníamos que llevar a nuestro AutorizaciónFilter. Bien. Entonces vamos a copiar con "Ctrl + S" y voy a comentar todas estas líneas, con doble barra.

[07:08] Doble barra, doble barra y doble barra. Hemos comentado todo, todo este código que nosotros vamos a llevar a nuestra AutorizaciónFilter, le vamos a colocar perdón que se colocó nuestra consola, se ejecutó el servidor porque se reinició, porque hemos hecho cambios en minimizar acá las cosas. Y bueno, volviendo a acá nuestro filter.

[07:38] Ese filter va a ejecutar nuestro código que ve la parte de la autorización. Bien, ya hemos pegado el código y ahora tenemos algunos problemas. Nos está diciendo que este get session no podemos obtener la sesión de nuestra request ¿Por qué no podemos obtener esa sesión de la request, si en nuestro servlet nosotros podíamos obtener esa sesión? ¿Qué es lo que ocurre?

[08:07] Recuerdan que yo les dije que todo servlet, o sea, este servlet que nosotros hemos creado recibe un `HttpServletRequest`. Ese `HttpServletRequest` es una versión más específica que tiene métodos específicos que nuestro `ServletRequest` que tenemos acá en nuestro filtro no tiene.

[08:30] Entonces tendríamos que convertir este `ServletRequest` que genérico a un `HttpServletRequest` que es específico y que sí tiene ese método. Vamos a hacer eso. Antes de hacer esa conversión, ese cast que vamos a hacer, vamos a cambiar ese `ServletRequest`, el nombre del parámetro, vamos a cambiarlo acá, en vez de llamarlo `request`, vamos a llamarlo `servletRequest`.

[08:56] Y este `ServletResponse` vamos a llamarlo `ServletResponse`. Bien, le hemos cambiado el nombre. Acá este déjenme que voy a hacer un enter en ese



throws para poder ver un poco mejor. Y perfecto, tenemos nuestro servletRequest. Ahora yo quiero obtener un HttpServletRequest. Entonces HttpServletRequest, está bien, puse bien, vamos a llamarlo de request, es igual a.

[09:28] Ahora tenemos que hacer el cast, convertir nuestro servletRequest en un HttpServletRequest. Vamos a hacerlo. Entre paréntesis decimos a cuál clase va a ser convertida. Va a ser convertida en HttpServletRequest y qué cosa va a ser convertida a eso. Nuestro servletRequest;

[09:52] Perfecto. Hemos hecho ese cast, podemos hacer ese cast porque como dijimos este HttpServletRequest, vean que tiene una extensión de ServletRequest, entonces podemos hacer ese cast, podemos hacer esa conversión. Así como lo hicimos con el request, vamos a hacerlo con el response, vamos a copiar esta línea. "Ctrl + C", "Ctrl + V" y en vez de un HttpServletRequest va a ser un ServletResponse.

[10:22] HttpServletResponse. Cambiamos request por response. El cast va a ser a response y serverResponse con R mayúscula. Okay, "Ctrl + S", "Ctrl + Shift + O" para importar que nos estaba dando unos errores y bien, ya se fueron algunos errores, nos queda este paramAccion. ¿Ese paramAccion lo podemos obtener de donde? Nosotros también lo obteníamos de acá, de nuestra request, HttpServletRequest.

[11:00] Entonces vamos a copiar ese código y vamos a colocarlo también acá. Bien, entonces tenemos nuestro paramAccion y nuestra sesión, bien. Ya en teoría estaría funcionando todo. Antes de hacer cualquier cosa, me gustaría colocar un system.out para poder saber que se ejecutó este filtro.

[11:28] Entonces voy a hacer un sysout, "Ctrl + Espacio" y ya me autocompleta y voy a colocar acá. Esto es autorización, voy a copiar el nombre ya que estoy, autorizaciónFilter, "Ctrl + C", "Ctrl + V" y voy a hacer lo mismo en

monitoreoFilter. Vamos a hacer lo mismo acá cuando se ejecute el doFilter, vamos a colocar un sysout.

[11:54] Y el nombre es monitoreoFilter. Bien, esto es para poder tener acá en la consola que nos diga cuál filtro se ha ejecutado. Perfecto, voy a dar un run en nuestro servidor por las dudas. Bien, nuestra consola está todo bien. Vamos a ir acá a nuestro listaEmpresas, voy a intentar acceder a listaEmpresas. Enter, y nos devuelve al login.

[12:22] Reiniciamos el servidor. Entonces nadie se ha logado, vamos a ver qué es lo que ocurre acá en nuestra consola. Tenemos un autorizaciónFilter, dos autorizaciones filter y un monitoreoFilter. ¿Por qué tenemos eso? Vamos a nuestra autorizaciónFilter y vean lo siguiente, nosotros acá intentamos primero entrar en lista empresas, entonces nuestra petición fue a listaEmpresas.

[12:48] La cuestión es que no teníamos autorización para entrar ahí porque reiniciamos el servidor, entonces estábamos deslogados. Entonces nos dice acá que bueno, como intentamos entrar en listaEmpresas y listaEmpresas no es ni login ni LoginForm, entonces nos envía a LoginForm, hace una redirección a LoginForm.

[13:10] Entró una nueva request para ir a LoginForm, entró de nuevo en autorización y nos dice: “ahora sí, LoginForm puede pasar”. Entonces de ahí fue al otro filtro, al monitoreo bien. Y ahí es donde se ejecutó toda esa parte de obtener el tiempo de ejecución, bien, entonces por eso tenemos esos dos autorizaciones y un monitoreoFilter.

[13:36] Ahora vamos a ver, vamos a intentar loguearnos, Bruno,12345, enviar, sí. Y vamos a hacer clic en modificar y vamos a ver qué más tenemos. Bien, tenemos acá una autorizaciónFilter, monitoreo, hicimos un login, tiempo de ejecución del Login 13 milisegundos. autorizaciónFilter de nuevo, monitoreo

de nuevo, lista las empresas y nos dice cuánto tiempo demoró en listar las empresas. Autorización, monitoreo y de nuevo ejecución.

[14:10] Entonces estamos viendo un patrón acá, estamos viendo que está siendo ejecutado primero un autorización y después un monitoreo, eso está bien, o sea, es lo que nosotros queremos, ¿pero cómo hacemos para definir ese orden? ¿Qué es lo que ocurriría si, por ejemplo, en el futuro decidimos convertir nuestro controlador en vez de un servlet que sea un filter por ejemplo? Podríamos hacerlo.

[14:38] Y ese controlador tiene que sí o sí estar al final, y nosotros acá en ningún lugar hemos decidido quién va a ir primero y quién va a ir después y quién va a ir último. Entonces, eso con nuestras anotaciones nosotros no podemos hacerlo, pero sí podríamos hacerlo usando nuestro web.xml. ¿Vieron que yo lo eliminé a nuestro filtro?

[15:04] Era por eso mismo, porque quería mostrarles esta situación. Qué es lo que ocurre, o sea, estas etiquetas nos dan cosas muy, muy buenas como poder hacer una cosa con una sola línea de código, pero a cambio no nos deja por ejemplo, en este caso definir el orden de ejecución de los filtros.

[15:20] Entonces, ya en el próximo video vamos a ver cómo hacer para resolver